CROSS EDUCATIONAL SOFTWARE
P. O. Box 1536
Ruston, Louisiana 71270
Phone 318-255-8921

PHYSICS
VOLUME 5

$15.00

CIRCULAR MOTION

CROSS
EDUCATIONAL
SOFTWARE

C I R C U L A R   M O T I O N

Steve Kamm
South Oklahoma City
Junior College
7777 South May
Oklahoma City, Oklahoma
73159

Mark Cross
Grambling State Univ.
Grambling, LA 71270

INTRODUCTION

Disk five in the physics series contains these programs:

CIRCLE 1 (constant speed)

CIRCLE 2 (acceleration)

SIMPLE HARMONIC MOTION 1 & 2

ORBIT THEORY

ROUND ORBIT GAME

DEMUFFIN

The two orbit programs were written by Mark Cross. The others were written by Steve Kamm with minor changes by Cross.

CONVERTING TO DOS 3.2

This diskette has data in the DOS 3.3 format (16 sectors). If your Apple has the older DOS 3.2 then the programs must be transferred to a blank DOS 3.2 diskette. Take the initialized blank and the new physics programs diskette to a computer that has DOS 3.3. Boot the physics diskette and then BRUN DEMUFFIN. When asked for the file name, type "=". This is a wild card that copies all files. Then DEMUFFIN gives more instructions and transfers files from DOS 3.3 down to the blank DOS 3.2 diskettes.

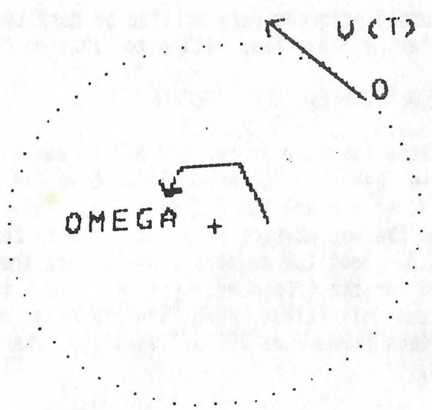
PROGRAM DESCRIPTIONS

CIRCLE 1 (constant speed)

This program explains the differences between angular velocity, tangential velocity, angular acceleration, tangential acceleration, and radial acceleration for circular motion with constant speed. It draws the picture below with the "O" object moving around the circle. Text appears on the screen to explain the different vectors and the picture is relabeled after each explanation.

The program concludes that if the speed of rotation is constant then these are true:

- Angular velocity is constant.
- Tangential velocity is not constant.
- Angular acceleration equals zero.
- Tangential acceleration equals zero.
- Radial acceleration equals V squared / r.



CIRCLE 2

Rotation with increasing speed

The program begins by showing a picture of accelerating circular motion. Then it asks the viewer whether the speed, angular velocity, and tangential velocity were constant. Next the program explains velocity and acceleration ending with:

```
---- tangential velocity     = radius * omega
---- tangential acceleration = radius * alpha
---- radial acceleration     = V squared / radius
```

Finally it asks the viewer these questions:

An electric fan is switched on. During the first few seconds, a point on the tip of a fan blade has:

1. Constant speed? (true or false)

2. Radial acceleration? (T or F)

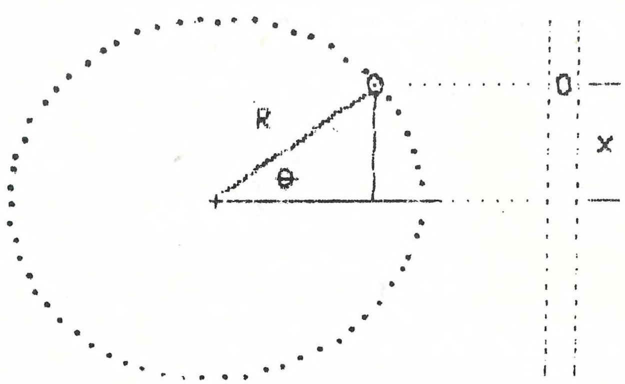
3. Constant angular velocity? (T or F)

4. No tangential acceleration? (T or F)

SIMPLE HARMONIC MOTION 1 & 2

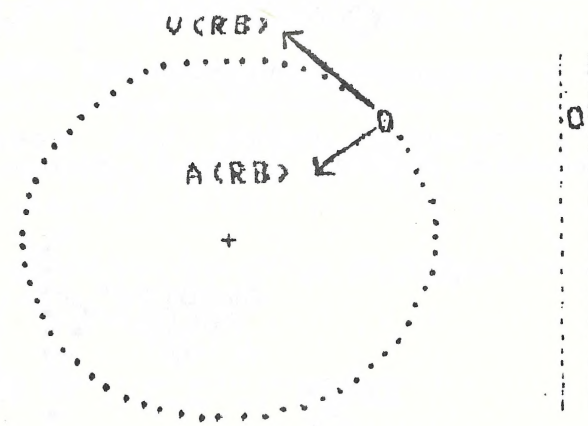
Another title for this program is "Simple Harmonic Motion and the Reference Circle." It often shows a picture of an object moving in a circle beside a projection of the same object moving vertically in simple harmonic motion. The program presents these steps for problem solving:

1. Pick out the mass, amplitude, and period or frequency from the given information.
2. The radius of the reference circle is equal to the amplitude of the simple harmonic motion.
3. The velocity of the rotating body equals circumference/period, where circumference is $2\pi r$.
4. Centripetal acceleration = V squared / radius.
5. Find the angle that corresponds to the displacement: theta = sin (SHM displacement/ radius).
6. SHM velocity = velocity of the rotating body * cos (θ).
7. SHM acceleration = acceleration of the rotating body times sin(θ).
8. Force on the SHM object = mass * acceleration.
9. Kinetic energy = (1/2) * mass * velocity squared.



Part 2 applies these steps to a detailed solution of this problem: A 1.5 kg mass oscillates 75 times per minute along a 10 cm. path. Find its velocity, acceleration, force, and kinetic energy when it is 3 cm from the center of oscillation.

The student is asked to solve the following problem by himself: A 16 pound weight oscillates along a 10 foot path with a frequency of two cycles per second. What is the force on the weight when it is two feet from the center? The computer waits for the student to finish, then presents its own answers for comparison.



ORBIT THEORY

What keeps gravity from pulling down a satellite? Physics has two answers. The correct explanation is that the satellite actually does fall, but it falls in a way that bends the motion from a straight line into a circle. This comes from Newton's laws.

The second answer uses forces and is a little easier for beginners to understand. It says that the outward centrifugal force exactly balances the pull of gravity.

BECAUSE OF NEWTON'S FIRST LAW
THE MOON WOULD NATURALLY MOVE
THROUGH SPACE IN A STRAIGHT LINE.

Earth's gravity makes the moon
fall toward Earth as it moves.

This program presents both explanations with pictures and, if the viewer wants, the equations that go with the centrifugal force explanation. The program shows the separate equations for gravity and centrifugal force, equates them, adn then does a little algebra to reduce them to a relation between velocity and radius in circular orbits. It concludes that a bigger orbit should have a slower speed.

The program teaches the concepts of gravity, centrifugal force, apogee, and perigee. A quiz at the end shows a satellite moving around Earth in an elliptical orbit. The computer asks the viewer to press any key when the speed is the fastest. The second question asks for a key to be pressed when gravity is weakest.

You can adjust the reading speed. Line 450 has a delay variable called SP = 41. Making this bigger will cause text to be written to the screen more slowly.

Q U I Z

NEXT QUESTION:

PRESS ANY KEY WHEN GRAVITY IS WEAKEST.

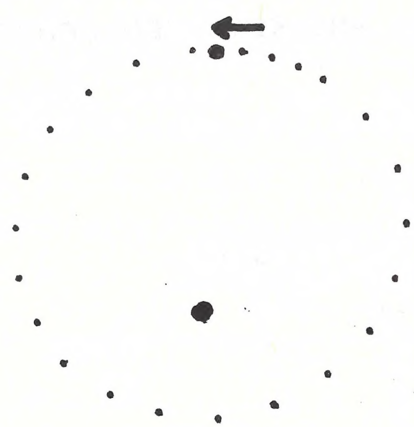
Gravity is weakest at the apogee.
YOU ARE 100 % CORRECT.

CIRCULAR ORBIT GAME

In the early days of space flight, news reports would draw attention to any mid-flight changes in the shape of the orbit. This program makes those changes into a game. The player is given an initial elliptical orbit. He has to change it into a circle using a limited amount of fuel. He can go faster or use retro-rockets using the F and R keys. He wins if he gets less than 10% difference between the apogee and perigee. There are several ways to lose: he can run out of fuel, run out of time, fly off the screen, or crash into the planet.

The game is very difficult for someone who doesn't know the physics secret. It can be solved with three rocket blasts by someone who understands the theory.

All orbits are periodic. This means that a ship will come back to wherever it is at the start of the orbit. Each rocket firing creates a new orbit, but this orbit includes the point where the rocket was fired. Thus the ship will come back to that spot. Therefore a blast can't change the height of the orbit at the place where it is fired. Each rocket blast must be calculated to affect the height on the OPPOSITE SIDE of the orbit.



PERIGEE = 42 APOGEE = 103

FUEL = 26

PROGRAMMING NOTES

CIRCULAR MOTION and SIMPLE HARMONIC MOTION

Every program on this disk has a shape table appended to the last line of the program. CIRCLE and SHM have a few letters used to label their high resolution pictures. Both programs use the HGR screen.

The CIRCLE and SIMPLE HARMONIC MOTION programs are organized like this:

Lines 100-999 are the main program. It flows straight through from line 130 to the end with few branches.

Lines 9000-9500 are special drawing subroutines, such as a circle or a Greek letter.

Lines 10000-10045 print the title.

Lines 10050-10070 are a subroutine that prints "RETURN" and waits for a carriage return.

The normal exit to "RUN MENU" is either at the end of the main program sequence or else in lines 15000-15100.

Line 63999 sets up the pointer to the beginning of the shape table.

PROGRAMMING NOTES

Orbit Theory and Round Orbit Game

ORBIT THEORY and ROUND ORBIT GAME have a shape table with the full set of ASCII characters appended after line 63999 of each program. The shapes are listed on page 12 of this manual.

ORBIT THEORY nearly fills up memory from 2048 to 16384, just below the second high resolution screen. If it is improved then the changes should not increase the size of the program above 16384 or else the program will destroy its own end when it first clears the HGR2 screen. The top of program pointer at PEEK (175) + 256 * PEEK (176) should be less than 16384.

MEMORY USAGE (ORBIT THEORY and ROUND ORBIT GAME)

```
---------------------------------------------------------------
Text screen |                  |HGR2 |                  |
            |     Program      |     |                  |
    and     |                  |     |    variables     |  DOS
            |   ( and shape    |screen|                 |
Monitor use |       table )    |     |                  |
---------------------------------------------------------------
$00        $0800              $4000  $6000             $9600
            2048              16384  24576
```


ORBIT THEORY and ROUND ORBIT GAME have a shape table about 1000 bytes long appended to the end of the last line of each Basic program:

```
63999 Z9 =  PEEK (121) + 256 *  PEEK (122):Z9 = Z9 + 73: POKE
      232,Z9 -  INT (Z9 / 256) * 256: POKE 233,Z9 / 256: RETURN

   - - -  A HIDDEN SHAPE TABLE FOLLOWS IN MEMORY  - - -
          91 SHAPES OF LETTERS, NUMBERS, ETC.
```

Subroutine 63999 sets up a pointer (Z9) to the start of the shape table. It is possible to delete all but line 63999 and then write a new program in front of it. So long as line 63999 is not disturbed the shapes will still be there.

The characters are all 5 wide x 7 high except the 2 x 2 box shape number one. Their origin is at the upper left of each shape. Draw 52 at 0,0 will put an R in the top left of the screen. Shape number 65 is a solid box used for erasing things under it.

Start
Drawing

The actual printing (drawing letters) is done by subroutine 10. It prints the string M$ starting at the cursor position CH (0-39) and CV (0-23). A printing flag, PF, tells the subroutine whether to print upper or lower case and whether to put a carriage return at the end. Example:

```
M$ = "TEST": PF = 0: CH = 18: CV = 0: GOSUB 10
```

will draw "TEST" in the top center of the screen in capital letters with a carriage return. PF = 0 or 3 makes a carriage return after printing. PF = 1 or 3 converts letters A-Z to lower case before printing.

SHAPE	NUMBER	ASCII NUMBER	SHAPE	NUMBER	ASCII NUMBER	SHAPE	NUMBER	ASCII NUMBER
BOX	1		>	32	62	]	62	93
SPACE	2	32	?	33	63	^	63	94
!	3	33	@	34	64	_	64	95
"	4	34	A	35	65	█	65	96
#	5	35	B	36	66	a	66	97
$	6	36	C	37	67	b	67	98
%	7	37	D	38	68	c	68	99
&	8	38	E	39	69	d	69	100
'	9	39	F	40	70	e	70	101
(	10	40	G	41	71	f	71	102
)	11	41	H	42	72	g	72	103
*	12	42	I	43	73	h	73	104
+	13	43	J	44	74	i	74	105
,	14	44	K	45	75	j	75	106
-	15	45	L	46	76	k	76	107
.	16	46	M	47	77	l	77	108
/	17	47	N	48	78	m	78	109
0	18	48	O	49	79	n	79	110
1	19	49	P	50	80	o	80	111
2	20	50	Q	51	81	p	81	112
3	21	51	R	52	82	q	82	113
4	22	52	S	53	83	r	83	114
5	23	53	T	54	84	s	84	115
6	24	54	U	55	85	t	85	116
7	25	55	V	56	86	u	86	117
8	26	56	W	57	87	v	87	118
9	27	57	X	58	88	w	88	119
:	28	58	Y	59	89	x	89	120
;	29	59	Z	60	90	y	90	121
<	30	60	[	61	91	z	91	122
=	31							

The characters are all 5 wide x 7 high except the 2 x 2 box shape number one. Their origin is at the upper left of each shape. Draw 52 at 0,0 will put an R in the top left of the screen. Shape number 65 is a solid box used for erasing things under it.

Start Drawing